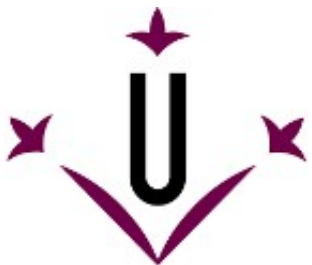


Advanced Programming in Artificial Intelligence

APAI Race

Grau en Enginyeria Informàtica
Universitat de Lleida

Josep Argelich



Context

- Practical case
- 35% of the total grade
- Minimum of 3 (over 10)
- In groups of 2 students (exceptionally 3)
- Design and development of a Local Search Solver for SAT + graph coloring + presentation of a research paper
- Final deadline (not the race): April 23th
- No participant in the race: Maximum grade of 7

The APAI Race: Submission

- Deadline: **April 13th, 11:00 AM (GMT +2)**
- Submission
 - Participants: Names of the group members
 - Solver: Name of the solver
 - Sent to the activity of the race in the Virtual Campus (with the names of the members of the group)
 - Benchmarks: 5 instances to upload into your shared space in the Virtual Campus (zip/tgz file)
 - File names: {id_group}_{id_instance}.cnf
 - Max size per instance: 100KB
 - Will be requested before the race

The APAI Race: Rules

- **Benchmarks**

- Files in CNF format (explained in class)
- Generated with the rnd-cnf-gen.py

- **Solver**

- Programming language: Python 3 (3.9.2 Feb 20th)
- Running parameters
 - \$./solver_name <input_cnf_formula>
- Output format
 - Comments (will be ignored): c comment
 - Solution: s SATISFIABLE
 - Values to variables: v -1 2 3 -4 5 -6 -7 -8 0

The APAI Race: Rules

- Benchmarks

```
c Random formula
p cnf 3 4
1 -2 3 0
2 3 -1 0
-1 -2 3 0
2 -3 0
```

- Solver output (the 'v' line can be another solution)

```
c Solver Name
s SATISFIABLE
v -1 2 3 0
```

The APAI Race: Resources

- Computational resources
 - Time: 10 seconds per instance (CPU time)
 - Memory: system limit
 - Be careful with swapping
- Penalization
 - Timeout: 2 x time per instance
 - Bug: 10000 seconds

The APAI Race: Results

- Results
 - Total time
 - CPU time for each solved instance
 - (2 x time per instance) for each timeout instance
 - 10000 seconds for each buggy solution
- Classification
 - Rank solvers by total time
 - Less (time) is better

The APAI Race: Environment

- Race environment
 - Give them **execution permission**: `chmod u+x <file>`
 - **race**: `$./race-incomplete.py <benchmark-folder> <solver>`
 - Runs every .cnf file in the benchmark folder with the solver
 - Sets the CPU time limit for each run
 - For each run outputs the CPU time or the penalization
 - At the end outputs the total time (score used to rank the solver)

After the Race

- If there is enough time...
 - Each group will explain the techniques and heuristics used in the solver
- Evaluation
 - Better solver → better grade for the practical case
 - The source code will be revised
 - May decrease the grade

Bonus track: Benchmarks

- **Graph Coloring Benchmarks**
 - Files in CNF format (explained in class)
 - Generated with the rnd-graph-gen.py
 - <num-nodes>: Number of nodes
 - <edge-prob>: Edge probability [0, 1]
 - <num-colors>: Number of colors to color the graph

Bonus track: example

- 3 nodes, 0.8 edge prob, 3 colors

c Random CNF formula

p cnf 9 21

1 2 3 0

-1 -2 0

-1 -3 0

-2 -3 0

4 5 6 0

-4 -5 0

-4 -6 0

-5 -6 0

7 8 9 0

-7 -8 0

-7 -9 0

-8 -9 0

-1 -4 0

-2 -5 0

-3 -6 0

-1 -7 0

-2 -8 0

-3 -9 0

-4 -7 0

c Solver Name

s SATISFIABLE

V -1 2 -3 4 -5 -6 -7 -8 9 0

Bonus track^{bis}

- Presentation of a research paper
 - Substitutes the first term exam
 - If you do this presentation, the grade will be the 15% belonging to the first term exam, so no need to do the mid term exam
 - Read and understand a research paper about local search in SAT
 - Make some slides explaining the paper
 - Make a 10-15 minutes presentation of the paper
 - In English and in streaming
 - Before the APAI Race day
 - Agree time and day

Advanced Programming in Artificial Intelligence

PAIA Race

Grau en Enginyeria Informàtica
Universitat de Lleida

Josep Argelich

